Lab Exercises

Supported Targets:

- PSoC6 BLE Pioneer Kit
- PSoC6 WIFI-BT Pioneer Kit
- CY8CPROTO-062-4343W

Software:

ModusToolbox™ 2.1

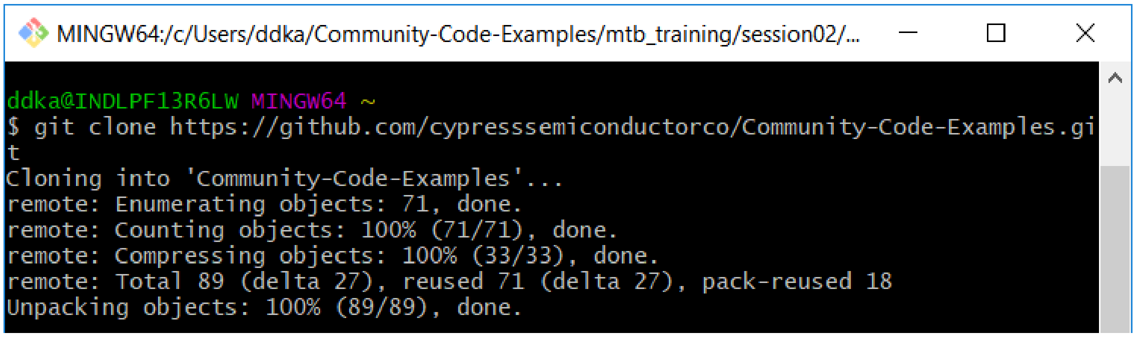
Repository Link:

https://github.com/cypresssemiconductorco/Community-Code-Examples/tree/master/mtb_training/session02

Importing the exercises:

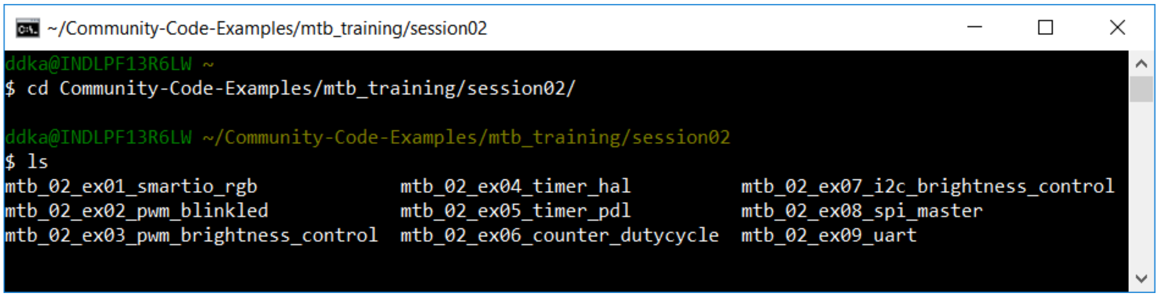
Here are the steps to import the projects into Eclipse IDE workspace

1. Clone the projects using the command:
   git clone https://github.com/cypresssemiconductorco/Community-Code-Examples.git


MINGW64:/c/Users/ddka/Community-Code-Examples/mtb_training/session02/...

```
ddka@INDLPF13R6LW MINGW64 ~
$ git clone https://github.com/cypresssemiconductorco/Community-Code-Examples.gi
t
Cloning into 'Community-Code-Examples'...
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (71/71), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 89 (delta 27), reused 71 (delta 27), pack-reused 18
Unpacking objects: 100% (89/89), done.
```

2. You will find all the exercises in the path *mtb_training/session02/* as shown:

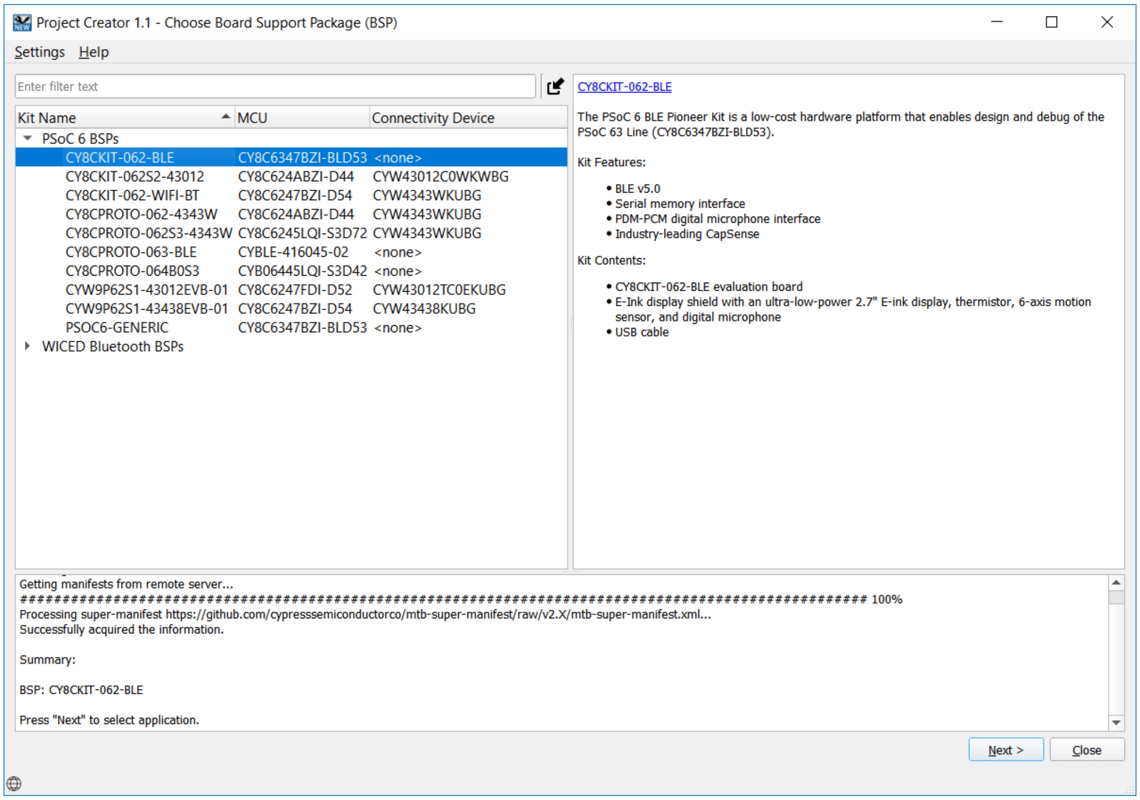


```
~/Community-Code-Examples/mtb_training/session02
ddka@INDLPF13R6LW ~
$ cd Community-Code-Examples/mtb_training/session02/

ddka@INDLPF13R6LW ~/Community-Code-Examples/mtb_training/session02
$ ls
mtb_02_ex01_smartio_rgb                mtb_02_ex04_timer_hal          mtb_02_ex07_i2c_brightness_control
mtb_02_ex02_pwm_blinkled               mtb_02_ex05_timer_pdl          mtb_02_ex08_spi_master
mtb_02_ex03_pwm_brightness_control     mtb_02_ex06_counter_dutycycle  mtb_02_ex09_uart
```

3. Open ModusToolbox and create a workspace folder of choice.

4. Once the Eclipse IDE is open, click **New Application > Choose BSP.**



Project Creator 1.1 - Choose Board Support Package (BSP)

Kit Name	MCU	Connectivity Device
PSoC 6 BSPs		
CY8CKIT-062-BLE	CY8C6347BZI-BLD53	<none>
CY8CKIT-062S2-43012	CY8C624ABZI-D44	CYW43012C0WKWBG
CY8CKIT-062-WIFI-BT	CY8C6247BZI-D54	CYW4343WKUBG
CY8CPROTO-062-4343W	CY8C624ABZI-D44	CYW4343WKUBG
CY8CPROTO-062S3-4343W	CY8C6245LQI-S3D72	CYW4343WKUBG
CY8CPROTO-063-BLE	CYBLE-416045-02	<none>
CY8CPROTO-064B0S3	CYB06445LQI-S3D42	<none>
CYW9P62S1-43012EVB-01	CY8C6247FDI-D52	CYW43012TC0EKUBG
CYW9P62S1-43438EVB-01	CY8C6247BZI-D54	CYW43438KUBG
PSOC6-GENERIC	CY8C6347BZI-BLD53	<none>
WICED Bluetooth BSPs		

CY8CKIT-062-BLE

The PSoC 6 BLE Pioneer Kit is a low-cost hardware platform that enables design and debug of the PSoC 63 Line (CY8C6347BZI-BLD53).

Kit Features:

- BLE v5.0
- Serial memory interface
- PDM-PCM digital microphone interface
- Industry-leading CapSense

Kit Contents:

- CY8CKIT-062-BLE evaluation board
- E-Ink display shield with an ultra-low-power 2.7" E-ink display, thermistor, 6-axis motion sensor, and digital microphone
- USB cable

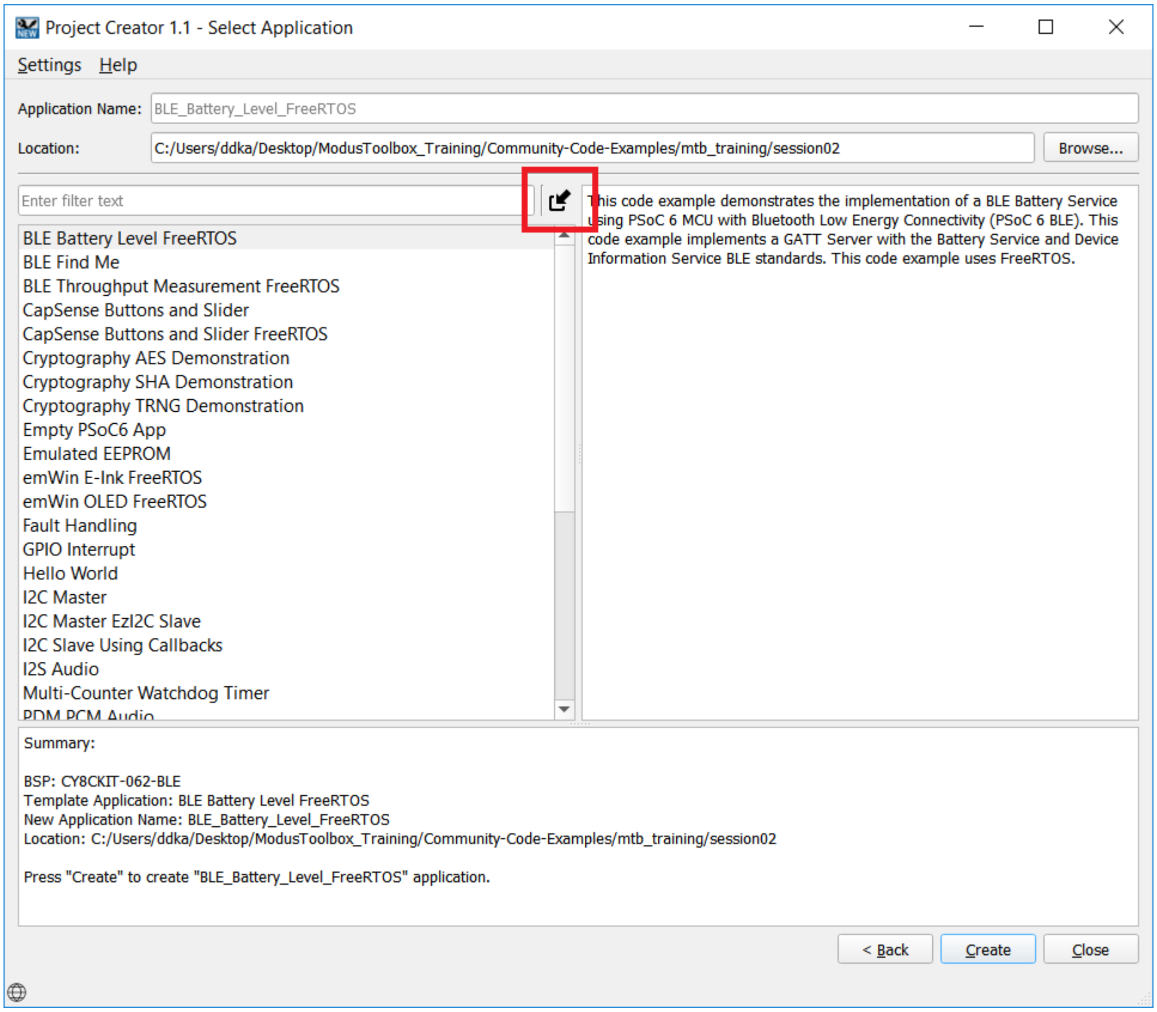
```
Getting manifests from remote server...
#################################################################################################### 100%
Processing super-manifest https://github.com/cypresssemiconductorco/mtb-super-manifest/raw/v2.X/mtb-super-manifest.xml...
Successfully acquired the information.

Summary:

BSP: CY8CKIT-062-BLE

Press "Next" to select application.
```

5. Click the Import symbol as shown below:



Application Name: BLE_Battery_Level_FreeRTOS

Location: C:/Users/ddka/Desktop/ModusToolbox_Training/Community-Code-Examples/mtb_training/session02

Enter filter text

- BLE Battery Level FreeRTOS
- BLE Find Me
- BLE Throughput Measurement FreeRTOS
- CapSense Buttons and Slider
- CapSense Buttons and Slider FreeRTOS
- Cryptography AES Demonstration
- Cryptography SHA Demonstration
- Cryptography TRNG Demonstration
- Empty PSoC6 App
- Emulated EEPROM
- emWin E-Ink FreeRTOS
- emWin OLED FreeRTOS
- Fault Handling
- GPIO Interrupt
- Hello World
- I2C Master
- I2C Master EzI2C Slave
- I2C Slave Using Callbacks
- I2S Audio
- Multi-Counter Watchdog Timer
- PDM PCM Audio

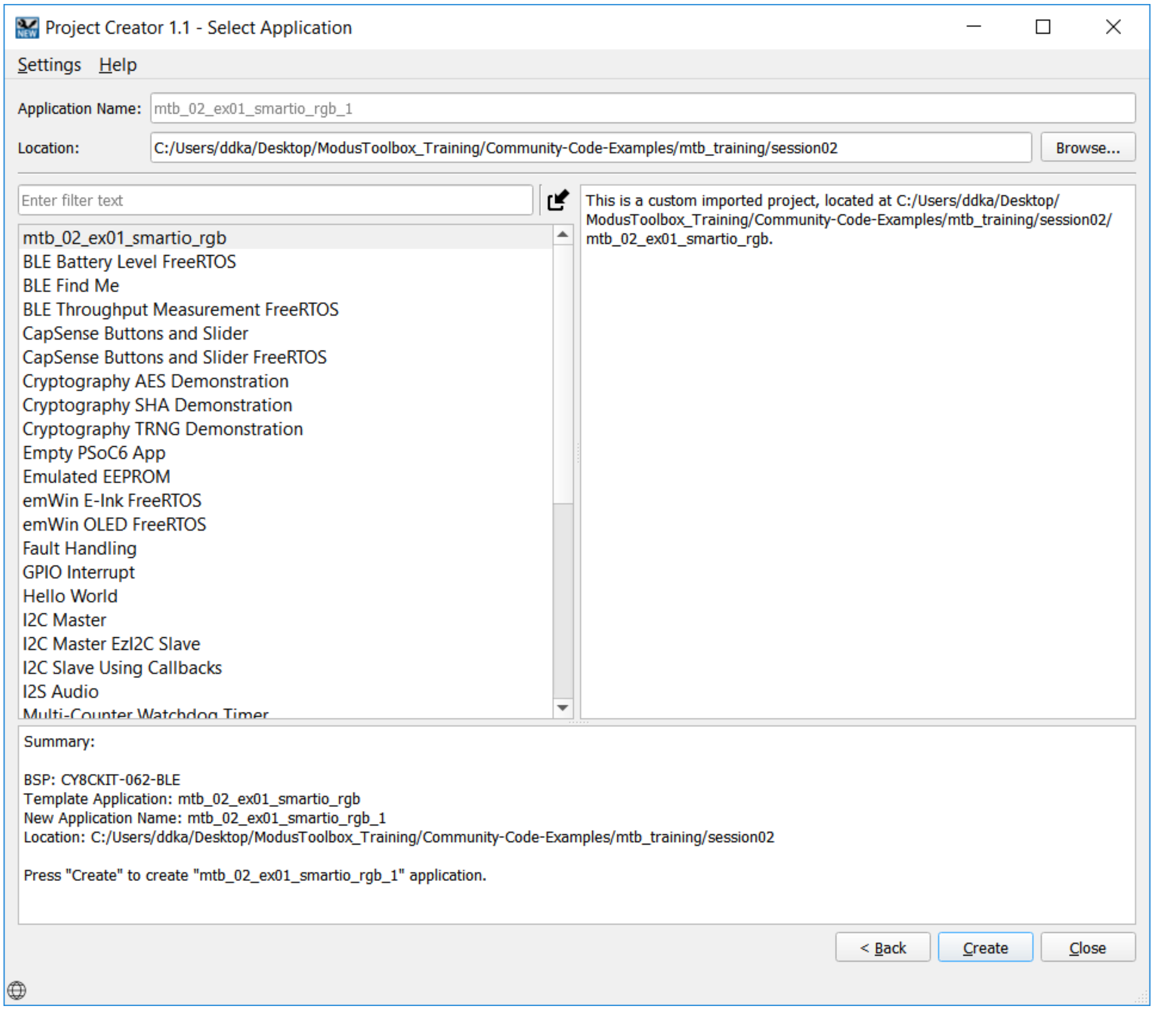
This code example demonstrates the implementation of a BLE Battery Service using PSoC 6 MCU with Bluetooth Low Energy Connectivity (PSoC 6 BLE). This code example implements a GATT Server with the Battery Service and Device Information Service BLE standards. This code example uses FreeRTOS.

Summary:

BSP: CY8CKIT-062-BLE
Template Application: BLE Battery Level FreeRTOS
New Application Name: BLE_Battery_Level_FreeRTOS
Location: C:/Users/ddka/Desktop/ModusToolbox_Training/Community-Code-Examples/mtb_training/session02

Press "Create" to create "BLE_Battery_Level_FreeRTOS" application.

6. Point to the directory of the exercise containing the Makefile. Once selected you should see the name of the exercise visible under Application Name as shown below:



Application Name: mtb_02_ex01_smartio_rgb_1

Location: C:/Users/ddka/Desktop/ModusToolbox_Training/Community-Code-Examples/mtb_training/session02

mtb_02_ex01_smartio_rgb
BLE Battery Level FreeRTOS
BLE Find Me
BLE Throughput Measurement FreeRTOS
CapSense Buttons and Slider
CapSense Buttons and Slider FreeRTOS
Cryptography AES Demonstration
Cryptography SHA Demonstration
Cryptography TRNG Demonstration
Empty PSoC6 App
Emulated EEPROM
emWin E-Ink FreeRTOS
emWin OLED FreeRTOS
Fault Handling
GPIO Interrupt
Hello World
I2C Master
I2C Master EzI2C Slave
I2C Slave Using Callbacks
I2S Audio
Multi-Counter Watchdog Timer

This is a custom imported project, located at C:/Users/ddka/Desktop/ModusToolbox_Training/Community-Code-Examples/mtb_training/session02/mtb_02_ex01_smartio_rgb.

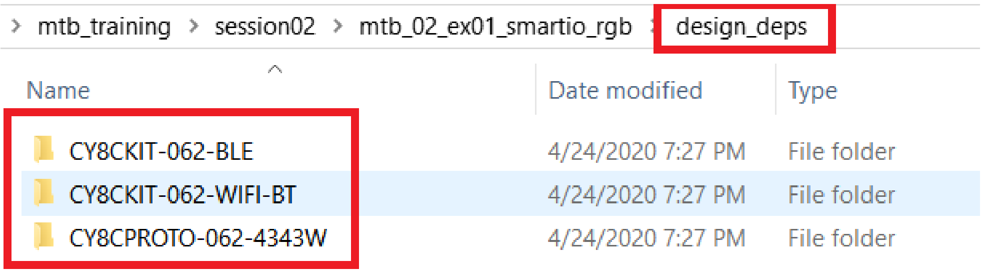
Summary:

BSP: CY8CKIT-062-BLE
Template Application: mtb_02_ex01_smartio_rgb
New Application Name: mtb_02_ex01_smartio_rgb_1
Location: C:/Users/ddka/Desktop/ModusToolbox_Training/Community-Code-Examples/mtb_training/session02

Press "Create" to create "mtb_02_ex01_smartio_rgb_1" application.

7. Click **Create** and then **Close**.

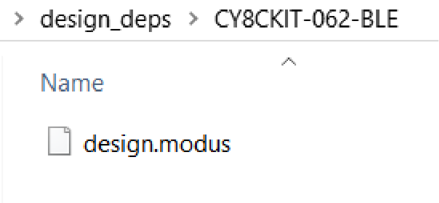
8. If the exercise you are working with uses PDL APIs, then it will contain a folder named "*design_deps*". It contains the supported target folders.



mtb_training > session02 > mtb_02_ex01_smartio_rgb > design_deps

Name	Date modified	Type
CY8CKIT-062-BLE	4/24/2020 7:27 PM	File folder
CY8CKIT-062-WIFI-BT	4/24/2020 7:27 PM	File folder
CY8CPROTO-062-4343W	4/24/2020 7:27 PM	File folder

Each of these targets will contain a *design.modus* file as shown below.

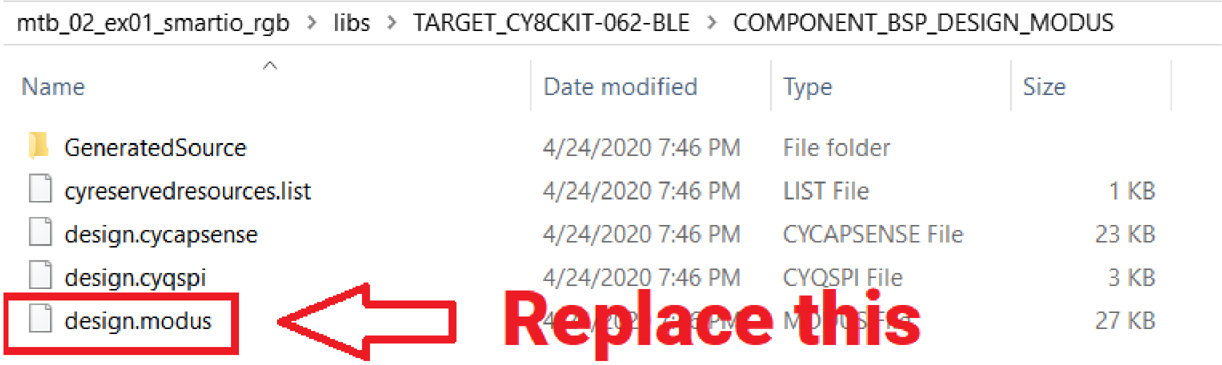


> design_deps > CY8CKIT-062-BLE

Name

design.modus

9. Copy the *design.modus* file of the target board you are working with and paste it into the following directory: *<exercise_name>\libs\TARGET_<BOARD_NAME>\COMPONENT_BSP_DESIGN_MODUS\*



mtb_02_ex01_smartio_rgb > libs > TARGET_CY8CKIT-062-BLE > COMPONENT_BSP_DESIGN_MODUS

Name	Date modified	Type	Size
GeneratedSource	4/24/2020 7:46 PM	File folder	
cyreservedresources.list	4/24/2020 7:46 PM	LIST File	1 KB
design.cycapsense	4/24/2020 7:46 PM	CYCAPSENSE File	23 KB
design.cyqspi	4/24/2020 7:46 PM	CYQSPI File	3 KB
design.modus	4/24/2020 7:46 PM	MODUS File	27 KB

Replace this

10. Delete the GenerateSource folder in the path: *<exercise_name>\libs\TARGET_<BOARD_NAME>\COMPONENT_BSP_DESIGN_MODUS\*

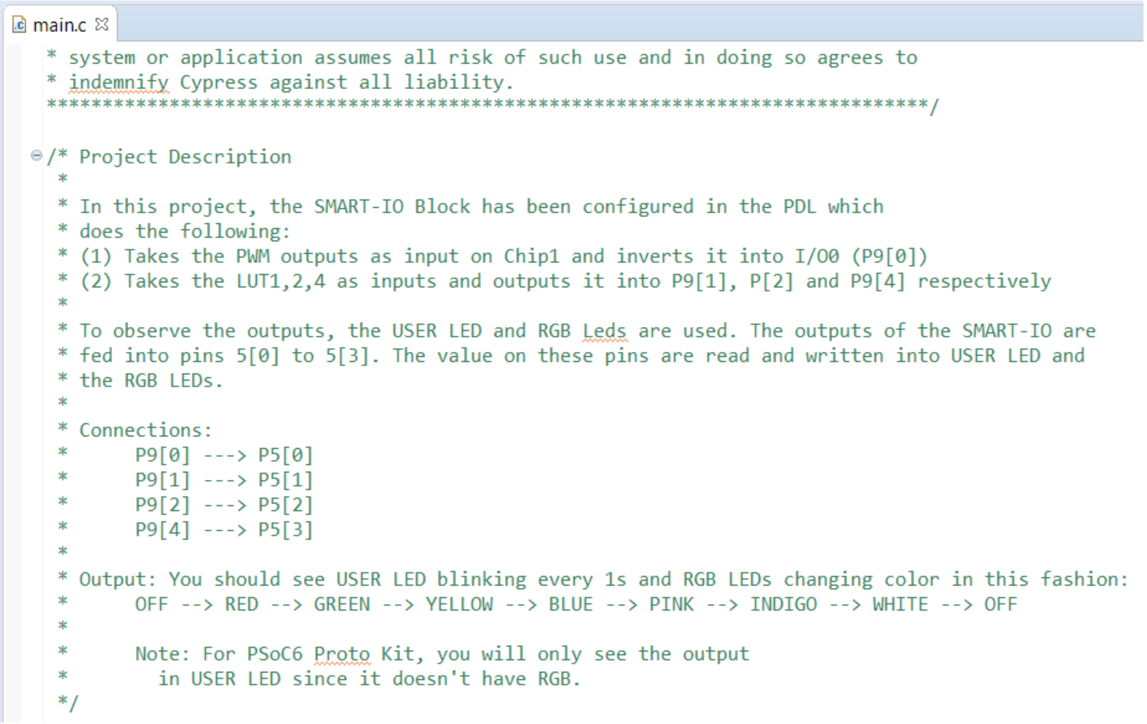
Running the exercises:

1. Open the Library Manager to choose the **Active BSP** and then click **Apply**.

2. Clean your application (mandatory step)

3. Build your application.

4. You can find the project description in the file *main.c* which explains the project and explains the pin connections and the expected output.



```
* system or application assumes all risk of such use and in doing so agrees to
* indemnify Cypress against all liability.
********************************************************************************/

/* Project Description
 *
 * In this project, the SMART-IO Block has been configured in the PDL which
 * does the following:
 * (1) Takes the PWM outputs as input on Chip1 and inverts it into I/O0 (P9[0])
 * (2) Takes the LUT1,2,4 as inputs and outputs it into P9[1], P[2] and P9[4] respectively
 *
 * To observe the outputs, the USER LED and RGB Leds are used. The outputs of the SMART-IO are
 * fed into pins 5[0] to 5[3]. The value on these pins are read and written into USER LED and
 * the RGB LEDs.
 *
 * Connections:
 *      P9[0] ---> P5[0]
 *      P9[1] ---> P5[1]
 *      P9[2] ---> P5[2]
 *      P9[4] ---> P5[3]
 *
 * Output: You should see USER LED blinking every 1s and RGB LEDs changing color in this fashion:
 *      OFF --> RED --> GREEN --> YELLOW --> BLUE --> PINK --> INDIGO --> WHITE --> OFF
 *
 *      Note: For PSoC6 Proto Kit, you will only see the output
 *        in USER LED since it doesn't have RGB.
 */
```

5. Program the device.